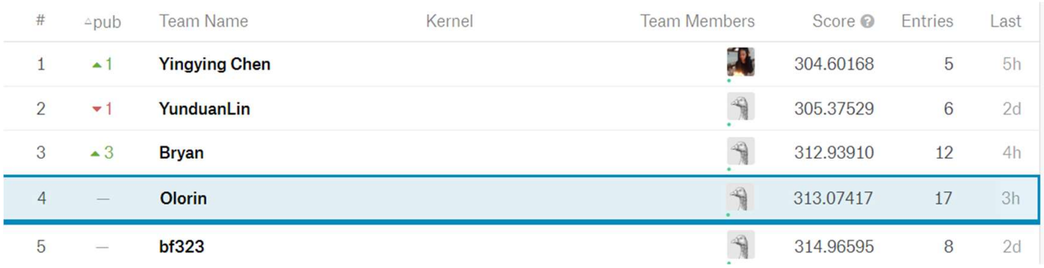
Assigment 4: Rank 4



#	∆pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲1	Yingying Chen			304.60168	5	5h
2	▼1	YunduanLin			305.37529	6	2d
3	▲3	Bryan			312.93910	12	4h
4	—	Olorin			313.07417	17	3h
5	—	bf323			314.96595	8	2d

1. Data Cleaning

The data contains 2012 Sept's SF data and 2015 whole year's NYC data.

1) NaN Value

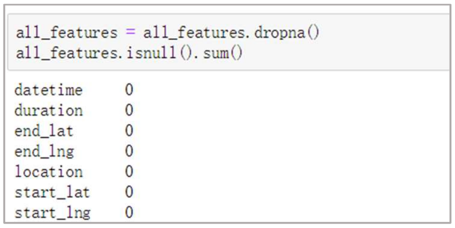
Some values are NaN in the dataset. Those are dropped.

2) Noise Data

- A bunch of data all have duration 40000, which is needed to be dropped from the dataset.
- Some of the data shows a weird speed pattern. They are dropped from the dataset as well.

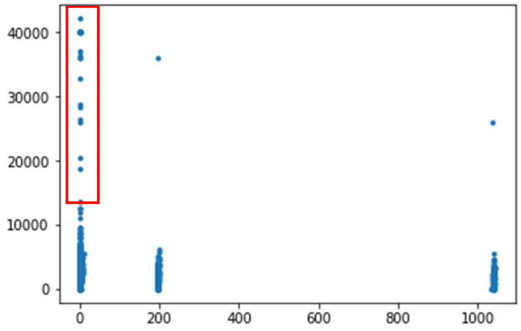
This is a draft of distance versus duration. Outliers could be observed.

- Speed is calculated for training set to do the data cleaning. All the speed that larger than 37 are dropped from the data set.



```
all_features = all_features.dropna()
all_features.isnull().sum()
```

```
datetime     0
duration     0
end_lat      0
end_lng      0
location     0
start_lat    0
start_lng    0
```



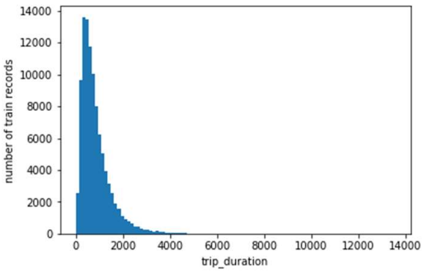
2. Exploratory data analysis

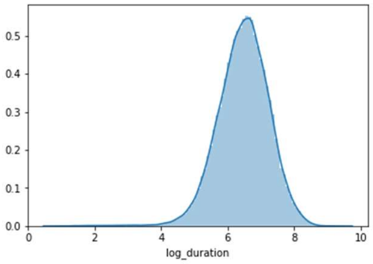
1) Distance and headings calculation

The distances are calculated as the distance between two points with the function in Assignment 2. Headings are calculated as azimuth and classified into 6 categories: N, S, NE, NW, SE, SW. They are made to dummy variables in the final model.

2) Normalization and Performance comparison

The duration is not normal distributed for both SF and NYC. This is the illustration from NYC data. I tried log the duration to make it normal distributed while the prediction results need to be logged back. The error could be created during the 'np.exp()' back part.





3) Clustering and Neighbors

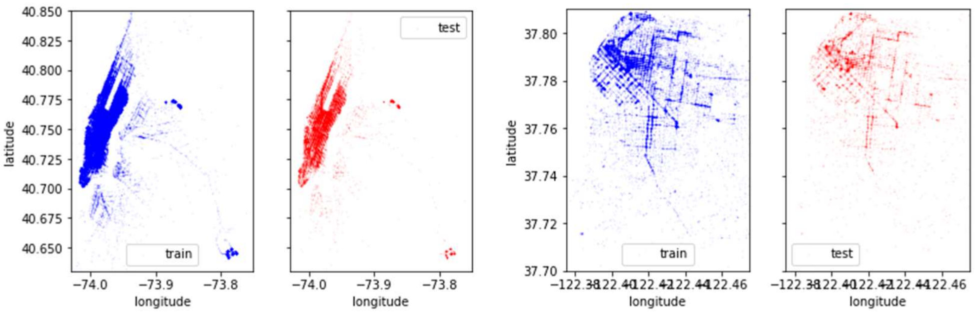
Neighbors are found as the length of a cluster. Using kmeans, each pick up location and drop off location is assigned to the number of neighbors they have.

4) PCA (2d to 2d)

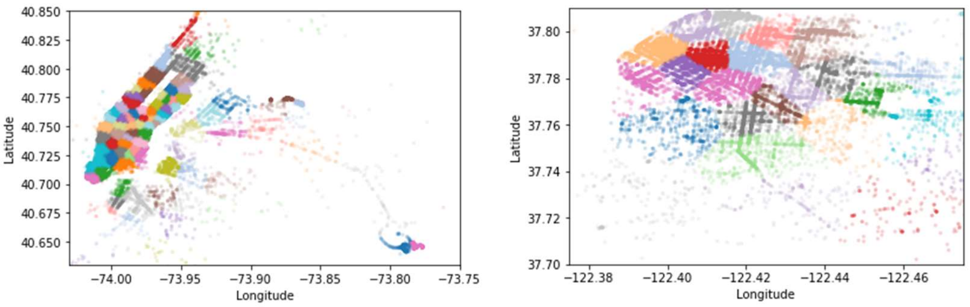
The PCA in this competition is not for the dimensional reduction. It is a 2D to 2D transformation. I find this way in NYC taxi data competition. It is good for tree growth and split according to the experiment.

5) Visualization

For the visualization, we could see that the dataset size for SF is relatively small. I tried training two separate models for the two cities, but the performance is not ideal. I think it is because of the dataset size. Therefore, I add a new feature which indicates which city the point belongs to. Data points



- Clusters



3. Modelling

Because of the page limitation, I only include the model. I started from neural networks which used to perform well in my past projects. However, this is not the case for this competition as the dataset and number of features is not big enough to build a deep neural network. Therefore, I turned my way to ensemble modelling and find XGBoost has the best performance.

1) Neural Network based on MXNet: 2-layer with one non-linear layer, bad result, around 380

2) Ensemble Method

- Random Forrest: Not so good, around 320
- Gradient Boosting: Not enough
- **XGBoost (chosen)**: 250 number of estimators and 10 max depth. Proved to be the best.

3） Concat Models

My best entries were 297 and 301, therefore I concat them together based on the final score. That is, the 297 model takes up for 301/598 and 301 model takes the other part. I make a new prediction based on the model. After the concatenation, the result is 294.